Transfer Learning Based Hybrid Quantum Neural Network Model for Surface Anomaly Detection

Sounak Bhowmik
Department of Electrical Engineering and Computer Science
University of Tennessee, Knoxville
Knoxville, USA
sbhowmi2@vols.utk.edu

Himanshu Thapliyal
Department of Electrical Engineering and Computer Science
University of Tennessee, Knoxville
Knoxville, USA
hthapliyal@utk.edu

***Abstract*—The rapid increase in the volume of data increased the size and complexity of the deep learning models. These models are now more resource-intensive and time-consuming for training than ever. This paper presents a quantum transfer learning (QTL) based approach to significantly reduce the number of parameters of the classical models without compromising their performance, sometimes even improving it. Reducing the number of parameters reduces energy consumption and training time and increases the models' flexibility and speed of response. For illustration, we have selected a surface anomaly detection problem to show that the resource-intensive and less flexible anomaly detection system (ADS) can be replaced by a quantum transfer learning-based hybrid model to address the frequent emergence of new anomalies better.**

***Index Terms*—quantum transfer learning, hybrid quantum neural networks, dressed quantum circuits, surface anomaly detection**

I. Introduction

Transfer learning [1] is a truly biologically inspired approach, applying knowledge gained from a specific context to another. Quantum Transfer Learning (QTL) [2], [3] is a branch of Quantum Machine Learning (QML) where we use the quantum mechanical properties offered by QML along with the knowledge transfer from benchmarked classical or quantum machine learning models. In this paper, we have applied QTL to reduce the number of parameters of a classical model, reducing its complexity without compromising its performance.

A. Motivation

With the advancement of artificial intelligence (AI), AI models have significantly increased in complexity in the past few decades [4]. We had a run for more sophisticated and powerful hardware to train bigger and more complex AI models. However, the more complex the model is, the more power it will consume for training and prediction. Moreover, training a complex model is time-consuming and difficult because of factors like bias, variance, and local optimum. Therefore, in this paper, we have tried to find out if reducing the number of parameters in a model without compromising its performance is possible.

Identify applicable funding agency here. If none, delete this.

For this purpose, we have picked a surface anomaly detection task and used quantum transfer learning (QTL) to approach this problem. QTL is a branch of Quantum machine learning (QML) that combines the power of machine learning algorithms and quantum computation. Properties like superposition, entanglement, and interference bring parallelism and speed-up in a quantum computer. QML models are lightweight, flexible, and less resource-intensive than the bulky deep learning models used in industrial applications. Therefore, an efficient, lightweight hybrid quantum model can be considered a replacement for some portion of a bulky classical model to reduce its total trainable parameters and computational complexity, reducing power usage and making them sustainable.

B. Contributions of the paper

In this paper, we proposed a classical-to-quantum transfer learning method for surface anomaly detection, which can be adopted easily in any other image-processing task. The contributions of this work are as follows,

- We first trained four classical deep neural network models on a binary classification task based on the NEU-DET [5] surface anomaly detection dataset. By random initialization, we ensured that none of the models suffered from local optimum.
- We applied quantum transfer learning to the classical models by replacing parts of the set of fully connected layers with a hybrid quantum neural network, which gave us sixteen total models: four QTL-based hybrid models based on each of the four classical models.
- Keeping the initial block of layers fixed, we trained only the replacement hybrid model through a 6-fold cross-validation process to assess their performance.
- From the performance data, we compare the performance and total number of parameters of the QTL-based hybrid models with their base classical model to show how well they perform even after reducing a significant number of parameters.

C. Organization of the paper

Section II presents the background of quantum transfer learning. Next, in section III, we described the workflow of

building a quantum transfer learning-based model. In section IV, we discuss the structural details of the classical and the hybrid quantum models. Section V discusses the dataset features, problem formulation, training parameters, and hardware specifications used in the experiment. In section VI, we show the simulation results of the transfer learning-based hybrid models and compare their performance and total number of parameters with the corresponding classical model. Finally, in section VII, we summarize the insights we gain from the experimental results and the future expectations from quantum transfer learning.

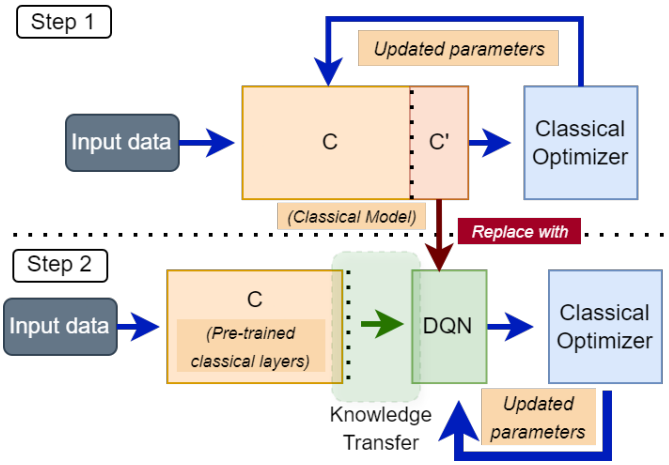


Fig. 1: Quantum Transfer Learning workflow.

II. Background

This section will discuss the concept of a variational quantum circuit and how it is applied to quantum transfer learning.

A. Variational Quantum Circuit

A Variational Quantum Circuit, or VQC, is a perfect analogy to a classical deep neural network. It consists of multiple layers of several single qubit rotational gates, followed by an array of entangling gates. The rotational gates are controlled by classical parameters, which can be tuned by any classical optimization method, such as gradient descent. A quantum layer can be defined as a unitary operation (U), which is applied on the input quantum state $|X_{n_q}\rangle$, consisting of n_q quantum subsystems or qubits, to produce an output state of $|Y_{n_q}\rangle$.

$$L : U(w)\,|X_{n_q}\rangle = |Y_{n_q}\rangle \tag{1}$$

Here, w is a set of classical trainable parameters.

To use a VQC to process classical features, we need to embed the classical features into the corresponding quantum states first. The embedding layer has a set of parameterized rotational gates, which, upon application on a basis state $|0^{\otimes n_q}\rangle$, controlled by the classical feature vector x_{n_q}, produces an embedded quantum state $|x^{\otimes n_q}\rangle$.

$$\mathcal{E} : U(x_{n_q})\,|0^{\otimes n_q}\rangle = |x^{\otimes n_q}\rangle \tag{2}$$

However, to get the classical output vector y, we need to measure the expectation values of n_q observables, $\hat{z} = [\hat{z_1}\hat{z_2}...\hat{z_{n_q}}]$, at the output. This measurement layer can be represented as,

$$M : |x^{\otimes n_q}\rangle \rightarrow y = \langle x^{\otimes n_q}|\hat{z}|x^{\otimes n_q}\rangle \tag{3}$$

Therefore, a VQC is a sequence of all these layers, viz.,

$$\mathbb{Q} = M \circ L \circ \mathcal{E} \tag{4}$$

B. Quantum Transfer Learning

Quantum transfer learning can be applied to a pre-trained classical deep learning model by replacing their last few dense layers with a VQC or a hybrid quantum model, where VQC has pre-processing and post-processing classical layers.

Assume C is a pre-trained classical model, and C' is the initial part of C, which consists of the layers specialized for feature extraction. Then, the transfer learning-based quantum model can be built as,

$$T = L_{n_q \rightarrow n_o} \circ \mathbb{Q} \circ L_{n_c \rightarrow n_q} \circ C' \tag{5}$$

Here, $\mathbb{Q}$ is a VQC consisting of n_q qubits, n_c is the dimension of the output vectors from C', and n_o is the number of output classes in the given problem. $L_{n_c \rightarrow n_q}$ is a preprocessing classical dense layer, *pre-net* producing output with a dimension of n_q. $L_{n_q \rightarrow n_o}$ is a postprocessing classical dense layer, *post-net*, that outputs a probability distribution over all the classes. The use of classical layers brings flexibility to the design of the VQC. That means the number of qubits used in the VQC does not need to depend on the output dimension of the pre-trained network or the number of output classes. In [3], the authors have named the set-up $L_{n_q \rightarrow n_o} \circ \mathbb{Q} \circ L_{n_c \rightarrow n_q}$ in equation 5, a dressed quantum network.

The total number of parameters in a dressed quantum network can be calculated according to equation 6. Here, n_{ip} is the input feature dimension of the *pre-net*, n_q = the number of qubits in the VQC, n_d = the number of fully entangled layers or the depth of the VQC, and n_c = the number of output classes.

$$\begin{aligned} W_{dqn} &= W_{pre-net} + W_{VQC} + W_{post-net} \\ W_{pre-net} &= n_{ip} * n_q + n_q \\ W_{VQC} &= 3 * n_q * n_d \\ W_{post-net} &= n_q * n_c + n_c \end{aligned} \tag{6}$$

In the upcoming sections, we will discuss the architecture of our transfer learning-based setup and compare its performance against the corresponding LeNet-based [6] classical models.

III. Workflow

There are two major steps in building a quantum transfer learning-based model.

- We first need to train a classical model. For that, we select a dataset, figure out a probable architecture for the classical model, and train it until we get an acceptable performance. In general, for image-processing applications, the models consist of an initial block of

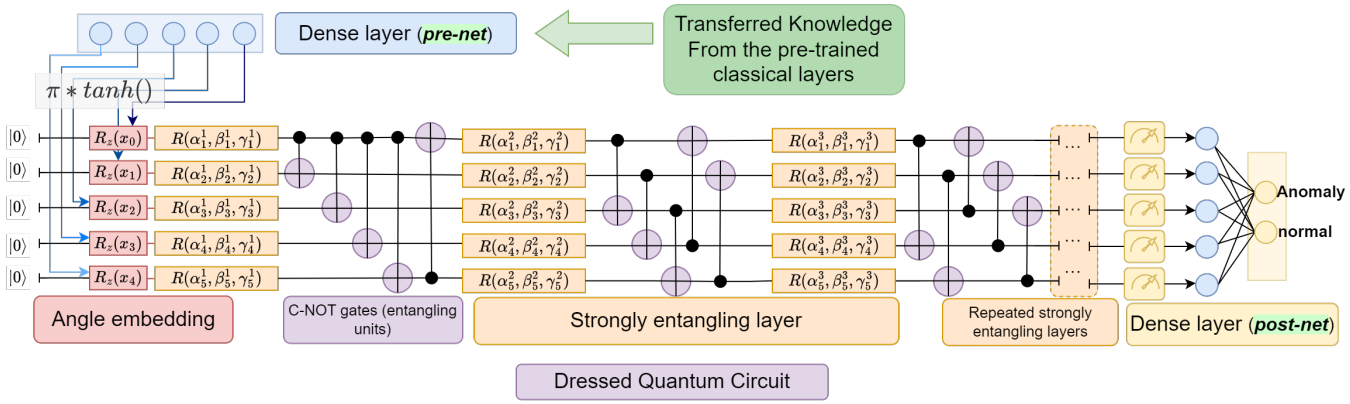


Fig. 2: The structure of a Dressed Quantum Network (DQN), which consists of a variational quantum circuit (VQC) between a pre-processing dense layer, pre-net, and a post-processing dense layer, post-net.

TABLE I: Architectures of different classical convolutional models used in this experiment

Model Name	Architecture	Number of Parameters
Model-1	Conv2D (1, 32, 32, 2) ∘ MaxPool2D (8, 1) ∘ Conv2D (32, 64, 16, 2) ∘ MaxPool2D (8, 1) ∘ Conv2D (64, 128, 16, 2) ∘ MaxPool2D (8, 1) ∘ Conv2D (128, 128, 2, 1) ∘ MaxPool2D (8, 2) ∘ Flatten () ∘ Linear (3200, 128) ∘ dropout (p=0.5) ∘ Linear (128, 64) ∘ dropout (p=0.25) ∘ Linear (64, 32) ∘ dropout (p=0.12) ∘ Linear (32, 16) ∘ Linear (16, 2)	**1,076,338**
Model-2	Conv2D (1, 32, 4, 2) ∘ MaxPool2D (4, 2) ∘ Conv2D (32, 64, 8, 2) ∘ MaxPool2D (2, 2) ∘ Conv2D (64, 128, 4, 2) ∘ Flatten () ∘ Linear (2048, 128) ∘ dropout (p=0.5) ∘ Linear (128, 64) ∘ dropout (p=0.25) ∘ Linear (64, 16) ∘ Linear (16, 2)	**534,482**
Model-3	Conv2D (1, 32, 8, 2) ∘ MaxPool2D (4, 2) ∘ Conv2D (32, 64, 4, 2) ∘ MaxPool2D (4, 1) ∘ Conv2D (64, 128, 2, 1) ∘ MaxPool2D (4, 2) ∘ Flatten () ∘ Linear (8192, 128) ∘ dropout (p=0.5) ∘ Linear (128, 64) ∘ dropout (p=0.25) ∘ Linear (64, 16) ∘ Linear (16, 2)	**1,125,842**

The general structure of the layers is as follows: Conv2D (input channels, output channels, kernel size, stride), MaxPool2D (kernel size, stride), Linear (input feature dimension, output feature dimension), dropout (p = the probability of an element being set to zero).

convolution and pooling layers with feature-extraction capability, followed by fully connected dense layers that classify the data based on the extracted features. In figure 1-step 1, these two blocks can be identified as C and C'.

- In the second step, we replace the second block of classical layers, C', with a dressed quantum circuit (DQN). To build the DQN, we chose the architecture of the variational quantum circuit (VQC) as our preference. There are two primary parameters to determine its structure, *viz.*, the number of qubits and the number of layers. Depending on the requirements and available resources, we can choose different configurations. For more information on the structure of VQC, refer to [7]. Finally, we need two more classical dense layers for pre-processing and post-processing data (pre-net and post-net, respectively). Refer to figure 2 for the detailed structure of the DQN we used in our experiment.

In the second step, we only update the parameters in the DQN, keeping the classical layers fixed at their pre-trained weights. This process ensures that we get the superior features extracted by the benchmark classical layers and then leverage the quantum mechanical properties of the DQN for feature fusion and classification. As we replace the bulky, dense layers with a relatively simple hybrid quantum-classical architecture, we can significantly reduce the total number of parameters of the working model.

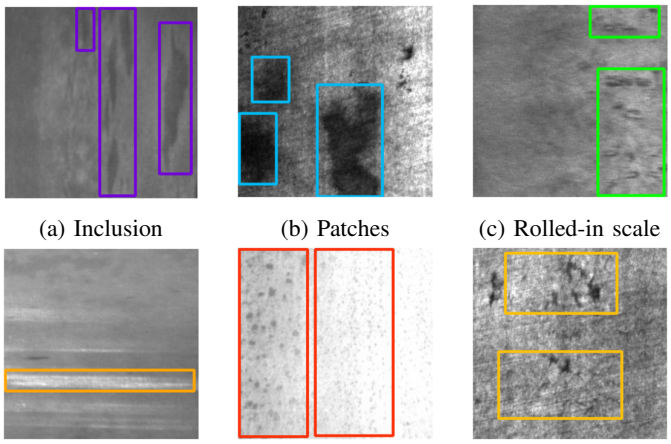
IV. Proposed Model

In this experiment, we have built three classical models using LeNet-based architecture. The details, such as the number and the dimension of the filters in the convolution layers, total number of layers, and number of units per layer, can be found in table I.

We also built four different configurations of quantum transfer learning-based models based on each classical model. We sequentially replaced the fully connected dense layers in our model, starting with one layer, then two, followed by three, and finally, all of them, using dressed quantum circuits in each case. Refer to figure 2 for the detailed architecture of the VQC, which consists of three parts.

- The first part, consisting of an array of single-qubit rotational gates, embeds the classical features extracted through the classical layers into quantum states. The rotational gates are controlled by the output of the *pre-net*, meaning that the rotation will depend on the classical features, and the output will be a quantum map of them.
- Next is a 3-layered, strongly entangled quantum circuit consisting of five qubits. These layers comprise a set of parameterized rotational gates and an array of C-NOT gates that entangle the qubits. The rotational gates of this layer are controlled by a set of classical parameters that we optimize by any classical optimizer, such as gradient descent. We have kept a low number of qubits and layers as the complexity rises exponentially with increasing depth. Figure 2 shows only one of the three strongly entangling layers of the VQC. The complete circuit has a cascaded structure with two more such layers.
- At the end, we measure the output in the Z-basis and get an expectation value used by the *post-net*.

The pre-net takes the extracted features from the pre-trained classical layers as an input, and its output dimension matches the number of qubits of the VQC. This process is important for the quantum embedding to work at the beginning of the VQC. The output of the post-net gives us the probability distribution over two classes: anomalous and normal. There are 45 trainable classical parameters in the VQC of every QTL-based hybrid model. But The number of parameters in *pre-net* varies according to the number of features extracted from the previous layers.



(a) Inclusion (b) Patches (c) Rolled-in scale

(d) Scratches (e) Pitted surface (f) Crazing

Fig. 3: Different types of surface anomalies in NEU-DET.

V. Experimental Set up

This section will discuss the used dataset, problem formulation, and training specification for the classical and hybrid working models.

A. Dataset (pre-processing and clean-up steps)

For this experiment, we have used the NEU-DET dataset, which has 300 images of each of the following six types of anomalies of the steel surface: *inclusion, patches, rolled-in scale, scratches, pitted surface, crazing*. Each image has a corresponding annotation file containing the location of the bounding box around the anomaly. Examples of each type of defect are shown in figure 3. All of the images are square-shaped RGB images of dimension 200x200. We initially made them gray-scale and standardized their value.

B. Problem formulation

In this work, we have formulated a binary classification problem, i.e., our classifiers can differentiate between an anomalous image and a normal one. As we did not have any normal images in the dataset, we had to generate a repository of normal surface patches by cropping the defect-free regions of the available images and resizing them to the original image dimension, 200x200. However, we saw many *pitted surface* and *crazing* type images had defects outside the annotated regions. So, we dropped these two types of defects. After clean-up, we had 1103 anomalous images, and we randomly selected 1103 normal images from our previously generated repository, which produced a total of 2206 images in our customized dataset.

TABLE II: Training hyperparameters and Hardware specs

Parameter	**Value**
Backbone	**LeNet-based classical layers**
Number of classes	**2**
Batch size	**64**
Image size	**1x200x200**
Optimiser	**Adam**
Learning rate	**0.001**
Loss	**Cross-entropy loss**
Activation functions	**soft-max (output), tanh(pre-net), ReLU(others)**
Epochs	**120 (Classical), 40 (Hybrid)**
Processor	**i7-10700 CPU @ 2.90GHz**
GPU	**NVIDIA Tesla T4**
RAM	**51 GB**
Quantum Simulator	**default.qubit, lightning.gpu (Pennylane)**

C. Training Specification

We initially trained four classical models on this data for 120 epochs, with a test size 20%. We initialized each model 5 times to avoid poor local minima problems. From them, we selected the best-performing model from each architecture. So, after this exhaustive search, we found three different models on which we apply quantum transfer learning in the later stage.

Then, We derived four QTL-based hybrid models from each classical architecture in the next step. Then, we performed 6-fold cross-validation for each of them. We recorded the results after 40 epochs for each fold. The training hyperparameters and the hardware used in this experiment are listed in Table II.

TABLE III: Performance and parameters comparison between the classical and their corresponding transfer-learning-based models

Model	F1 score(%)	Boost Perf(%)	#Params	Reduc. #Params (%)
CM-1	88.58	-	1,076,338	-
QTL-M-2	88.91	0.38	1,074,078	0.21
QTL-M-3	88.62	0.05	1,066,142	0.95
QTL-M-4	90.08	1.69	671,764	37.59
CM-2	95.16	-	534,482	-
QTL-M-2	96.62	1.54	533,790	0.13
QTL-M-3	96.81	1.73	525,824	1.62
QTL-M-4	98.95	3.98	273,182	48.89
CM-3	97.32	-	1,125,842	-
QTL-M-2	97.81	0.50	1,125,150	0.06
QTL-M-3	97.90	0.60	1,117,214	0.77
QTL-M-4	97.85	0.55	108,830	90.33

The information of the four configurations of quantum transfer learning-based models based on each classical model is depicted following the classical model-1 to 3 and named as Quantum Transfer Learning-based Model-2 to 4.

VI. Results

The classical models have been trained and validated on the same training and validation set. While training the classical models, we recorded the test accuracy after every epoch. The best performance from the classical model 2. At the end of 120 training epochs, it achieved an F1 score of 0.9732 and a test accuracy of 97.28%.

In the next step, during the 6-fold cross-validation process of the QTL-based hybrid models, we recorded the test loss and test accuracy after every epoch for each fold. To show the convergence of these models, we chose the fold with the best results in figures 4, 5, and 6. We have normalized the test losses betwee 0 and 2 for better visualization.

In table III, we have compared the performance and total number of parameters of the QTL-based hybrid models with their corresponding classical models. We chose the F1 score as the performance metric as it is a better indicator for a binary classifier. The table shows that the QTL-based model-4 for each classical model significantly reduces the total number of parameters without compromising their performance. There are reductions by 38%, 49%, and 90% for this model. The parameter reduction can be calculated according to equation 7, where W_i is the total number of parameters in ith classical

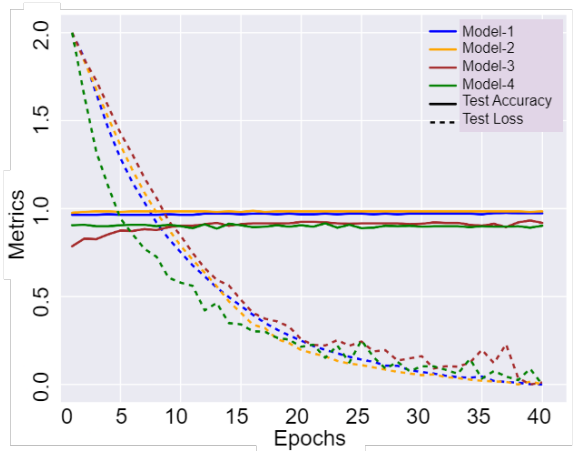


Fig. 4: Convergence plot: Transfer learning-based models, with the last *two* dense layers substituted by dressed quantum network.

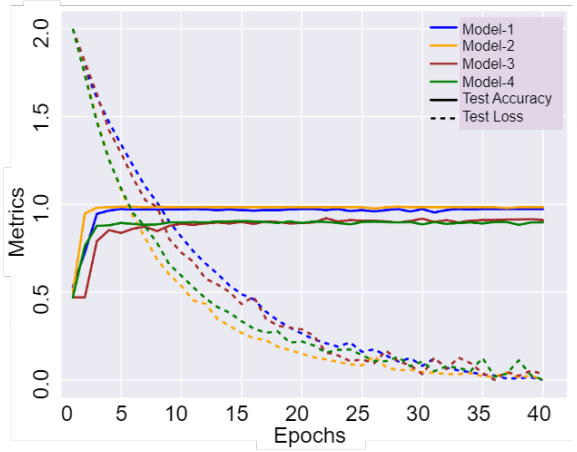


Fig. 5: Convergence plot: Transfer learning-based models, with the last *three* dense layers substituted by dressed quantum network.

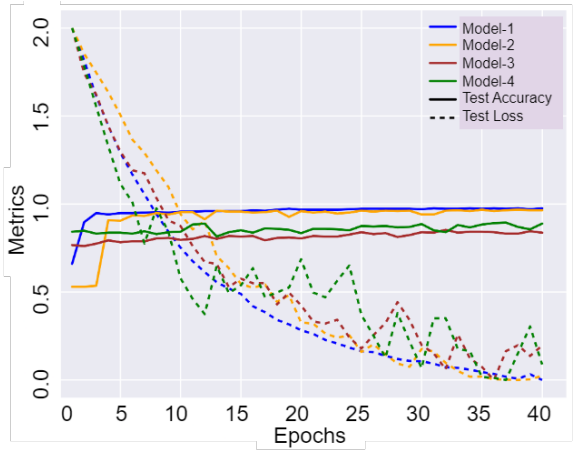


Fig. 6: Convergence plot: Transfer learning-based models, with *all but the convolutional* layers substituted with dressed quantum network.

layer and x is the number of replaced layers. W_{dqn} is the number of parameters in the dressed quantum circuit, calculated according to equation 6.

$$R = \frac{\sum_{i=1}^{x} W_i - W_{dqn}}{\sum_{i=1}^{x} W_i} * 100\% \tag{7}$$

However, the hybrid configurations 1 to 3 did not appreciate much parameter reduction. There is a good reason for that. Following the block of convolution and pooling, the first dense layer comprises the maximum number of parameters. We replace this layer for all the models in the QTL-based model-4 configuration only. That is why there is a huge reduction in the number of parameters in the fourth configuration and a very small reduction for the rest.

Following table III, we can see that, besides the reduction in the total number of parameters, the F1 scores of the quantum transfer learning-based models have also increased compared to the base classical model. Therefore, in a tradeoff between performance and model complexity, we saw that without any compromise in the performance, we could significantly reduce the complexity, *i.e.*, the total number of parameters in a classical model using a quantum transfer learning-based hybrid approach.

VII. Conclusion

In this experiment, we built four classical models. We applied quantum transfer learning, replacing different blocks of fully connected layers with a hybrid quantum circuit, keeping the rest of the pre-trained classical layers fixed, and training only the quantum circuit. In this process, we could significantly reduce the total number of parameters in the models and improve their performance with few exceptions. This phenomenon can be explained in light of the efficient feature extraction capability of the classical convolution layers and the quantum circuit's feature fusion and mapping capability due to their unique quantum mechanical properties.

Reduction in the total number of parameters will reduce the complexity and power usage, making the models more sustainable and easy to train. Lightweight quantum models are more flexible; therefore, they can easily be modified on demand.

In the current NISQ era, we can best utilize the power of quantum machine learning models with efficient hybrid approaches. This mitigates the hardware limitations, such as a limited number of qubits and noisy quantum subsystems. On top of that, we can leverage the power of quantum processing to empower the classical benchmark models.

Authors and Affiliations

Acknowledgment

References

[1] L. Torrey and J. Shavlik, "Transfer learning," in *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*. IGI global, 2010, pp. 242–264.

[2] L. Wang, Y. Sun, and X. Zhang, "Quantum deep transfer learning," *New Journal of Physics*, vol. 23, no. 10, p. 103010, 2021.

[3] A. Mari, T. R. Bromley, J. Izaac, M. Schuld, and N. Killoran, "Transfer learning in hybrid classical-quantum neural networks," *Quantum*, vol. 4, p. 340, 2020.

[4] G. Menghani, "Efficient deep learning: A survey on making deep learning models smaller, faster, and better," *ACM Computing Surveys*, vol. 55, no. 12, pp. 1–37, 2023.

[5] K. Song and Y. Yan, "A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects," *Applied Surface Science*, vol. 285, pp. 858–864, 2013.

[6] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, pp. 2278–2324, 1998. [Online]. Available: https://api.semanticscholar.org/CorpusID:14542261

[7] S. Sim, P. D. Johnson, and A. Aspuru-Guzik, "Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms," *Advanced Quantum Technologies*, vol. 2, no. 12, p. 1900070, 2019.